

(43) Date of publication of application: **08.08.87**

H03M 13/00

(71) Applicant: **SONY CORP**

(72) Inventor: **SAKO YOICHIRO**  
**YAMAMURA SHINICHI**

inverting processings as above to a prescribed number or below.

COPYRIGHT: (C)1987,JPO&Japio

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭62-181530

⑬ Int. Cl.<sup>4</sup>

識別記号

庁内整理番号

⑭ 公開 昭和62年(1987)8月8日

H 03 M 13/00

6832-5J

審査請求 未請求 発明の数 1 (全7頁)

⑮ 発明の名称 エラー訂正符号の復号方法

⑯ 特 願 昭61-23540

⑰ 出 願 昭61(1986)2月5日

⑱ 発 明 者 佐 古 曜 一 郎 東京都品川区北品川6丁目7番35号 ソニー株式会社内  
⑲ 発 明 者 山 村 真 一 東京都品川区北品川6丁目7番35号 ソニー株式会社内  
⑳ 出 願 人 ソニー株式会社 東京都品川区北品川6丁目7番35号  
㉑ 代 理 人 弁理士 杉浦 正知

明 細 書

1. 発明の名称

エラー訂正符号の復号方法

2. 特許請求の範囲

デジタルデータの複数シンボルの2次元配列の第1の方向に整列する複数個の上記シンボルからなる第1の系列の夫々に第1のエラー訂正符号の符号化がされ、上記2次元配列の第2の方向に整列する複数個の上記シンボルからなる第2の系列の夫々に第2のエラー訂正符号の符号化がされるエラー訂正符号の復号方法において、

上記第1の系列及び上記第2の系列が交互に処理されるように、上記設定された系列から隣接する系列に上記2次元配列の系列の訂正処理を順次行うステップと

上記順次なされる訂正処理の間で、エラー訂正が不可能な場合に、上記第1の系列の訂正処理と上記第2の系列の訂正処理との順序を1度反転させる処理を行うステップと、

上記反転させる処理の回数を所定数以下に制限

するステップと

からなることを特徴とするエラー訂正符号の復号方法。

3. 発明の詳細な説明

〔産業上の利用分野〕

この発明は、種符号、クロスインターリーブ符号等のエラー訂正符号の復号方法に関する。

〔発明の概要〕

この発明は、デジタルデータの複数シンボルの2次元配列の互いに異なる方向に整列する第1の系列及び第2の系列の夫々に第1のエラー訂正符号及び第2のエラー訂正符号の符号化がされるエラー訂正符号の復号方法において、所定の第1の系列又は所定の第2の系列の一方の系列を最初に訂正処理を行う系列に設定し、次に、第1の系列及び第2の系列が交互に処理されるように、設定された系列から隣接する系列に2次元配列の系列の訂正処理を順次行い、この順次なされる訂正処理の間で、エラー訂正が不可能な場合に、第1

の系列及び第2の系列の訂正処理の順序を1度反転させる処理を行い、この反転処理の回数を所定数以下に制限することにより、所定の演算時間内でのエラー訂正能力の向上を図るもので、特に、バーストエラーを効果的に訂正できるようにしたものである。

#### 〔従来の技術〕

光磁気ディスクにデジタルデータを記憶する場合に、データの書き込み又は読み出し時に生じるエラーを訂正するために、積符号が用いられる。積符号は、デジタルデータの2次元配列（マトリクスブロック）の各列及び各行に関して、エラー訂正符号の符号化を行うもので、エラー訂正符号としては、線形符号が用いられる。

従来の積符号の復号方法では、例えば特開昭60-116230号公報に示されるように、列方向のエラー訂正符号C1の復号を符号C1の全ての系列について行うC1復号と行方向のエラー訂正符号C2の復号を符号C2の全ての系列につい

て行うC2復号とが交互になされている。

#### 〔発明が解決しようとする問題点〕

積符号のマトリクスブロックの斜め方向（対角線方向）にデータを伝送する時に発生するバーストエラーを訂正する場合、従来の復号方法は、符号C1の復号と符号C2の復号の夫々の繰り返し回数を多くしないと、エラーシンボルを訂正することができない問題点があった。従って、積符号の1個のマトリクスブロックの復号に割り当てられている演算時間が十分に長くない時には、訂正されないエラーシンボルが残る問題があった。

上述の問題は、積符号に限らずクロスインターリーブ符号に関しても同様に発生する。クロスインターリーブ符号は、デジタルデータの2次元配列の異なる2つの斜め方向に並ぶ系列毎に第1のエラー訂正符号及び第2のエラー訂正符号の符号化を行い、列方向がデータ伝送方向とされたものである。従来のように、第1のエラー訂正符号の復号を全ての符号系列について行い、次に第2

3

のエラー訂正符号の復号を全ての符号系列について行う復号方法は、バーストエラーを効果的に訂正することができない。

従って、この発明の目的は、バーストエラーを効果的に補正することができるエラー訂正符号の復号方法を提供することにある。

#### 〔問題点を解決するための手段〕

この発明は、デジタルデータの複数シンボルの2次元配列の第1の方向に整列する複数個のシンボルからなる第1の系列C1<sub>1</sub>、C1<sub>2</sub>、・・・、C1<sub>n</sub>の夫々に第1のエラー訂正符号C1の符号化がされ、2次元配列の第2の方向に整列する複数個のシンボルからなる第2の系列C2<sub>1</sub>、C2<sub>2</sub>、・・・、C2<sub>m</sub>の夫々に第2のエラー訂正符号C2の符号化がされるエラー訂正符号の復号方法において、所定の第1の系列C1、又は所定の第2の系列C2、の一方の系列を最初に訂正処理を行う系列に設定するステップと、第1の系列及び第2の系列が交互に処理されるように、設定された系列

5

から隣接する系列に2次元配列の系列の訂正処理を順次行うステップと、この順次なされる訂正処理の間でエラー訂正が不可能な時に、第1の系列の訂正処理と第2の系列の訂正処理との順序を1度反転させる処理を行うステップと、この反転させる処理の回数を所定数以下に制限するステップとからなることを特徴とするエラー訂正符号の復号方法である。

#### 〔作用〕

デジタルデータの複数シンボルの2次元配列（マトリクスブロック）の列方向にエラー訂正符号C1の符号化がされ、その行方向にエラー訂正符号C2の符号化がされている積符号の場合、列方向の系列である列ブロックC1<sub>1</sub>、C1<sub>2</sub>、・・・、C1<sub>n</sub>と行方向の系列である行ブロックC2<sub>1</sub>、C2<sub>2</sub>、・・・、C2<sub>m</sub>とが1ブロック毎に交互に訂正処理される。マトリクスブロックの斜め方向にデータを伝送する場合、バーストエラーによって、1個の列ブロック又は1個の行ブロックにエラー

6

訂正可能な個数例えば1個のエラーシンボルより多い2個以上のエラーシンボルが含まれるエラーパターンが発生する。このようなエラーパターンの場合に、所定の列ブロック又は所定の行ブロックから隣接するブロック毎に列ブロック及び行ブロックを交互に処理することにより、エラーシンボルが次々と訂正される。従って、所定の時間内で復号処理を行う場合に、エラー訂正能力を実質的に上げることができる。

また、エラー訂正が不可能な時に、列ブロックと行ブロックとの訂正処理を1度反転させることにより、例えば(C1, -C2)の順序で処理する時に、列ブロックC1に含まれるエラーの訂正が不可能であり、行ブロックC2に含まれるエラーの訂正が可能であると、(C2, -C1)のように訂正処理を反転させる。これによって、バーストエラーの訂正できる能力のより向上を図ることができる。

#### (実施例)

生回路2がメモリ1に対して列ブロックのデータを読み出すためのアドレスデータを発生する時には、メモリ1とC1復号器4との間が接続され、読み出された列ブロックのデータがC1復号器4に供給され、訂正処理がこの列ブロックに関してなされ、訂正がされた列ブロックのデータがメモリ1に書き込まれる。一方、アドレス発生回路2がメモリ1に対して行ブロックのデータを読み出すためのアドレスデータを発生する時には、メモリ1とC2復号器5との間が接続され、読み出された行ブロックのデータがC2復号器5に供給され、訂正処理がこの行ブロックに関してなされ、訂正がされた行ブロックのデータがメモリ1に書き込まれる。

アドレス発生回路2は、復号開始の列ブロック又は行ブロックから列ブロックのデータと行ブロックのデータとを交互にメモリ1から読み出すようなアドレスデータを発生する。復号開始のスタートブロックは、制御回路6によって設定される。この一実施例では、所定の列ブロックC1。又は

以下、この発明を積符号に適用した一実施例について図面を参照して説明する。第1図は、この発明による復号方法を実施するための復号装置を示す。

第1図において、1で示すメモリ(RAM)に積符号のマトリクスブロックを構成するデジタルデータ及びチェックシンボルが記憶されている。メモリ1に記憶されているデータは、光磁気ディスク(図示せず)から再生された1セクターのデータである。メモリ1の書き込みアドレス及び読み出しアドレスは、アドレス発生回路2により生成される。メモリ1から読み出された1個の列ブロック或いは行ブロックのデータは、入出力制御回路3に供給される。

入出力制御回路3には、列方向のエラー訂正符号C1の復号器(C1復号器)4と行方向のエラー訂正符号C2の復号器(C2復号器)5とが接続されている。アドレス発生回路2及び入出力制御回路3は、制御回路6から発生する制御信号によって制御される。制御回路6によりアドレス発

所定の行ブロックC2。の一方がスタートブロックに設定される。

また、制御回路6は、列ブロックと行ブロックとを交互に訂正処理する時に、エラー訂正が不可能な時に列ブロックと行ブロックとの訂正処理を1度反転させるための制御信号を発生し、この制御信号がアドレス発生回路2に供給される。このため、C1復号器4及びC2復号器5から訂正が可能かどうかを示す判別信号が制御回路6に供給される。この反転処理の回数は、復号時間が長くなることを防止するために所定数(a)以下に制限される。aは、例えば1に設定される。

第2図は、この発明を適用することができる積符号の一例の構成を示す。第2図に示すように、M行、N列のマトリクス状に配列された(M・N)個のシンボルからなるマトリクスブロックによって、符号化の単位が形成される。[(M-P) × (N-Q)]個のデジタルデータのシンボル(例えば1シンボルが1バイト)の列ブロック毎及びその行ブロック毎にエラー訂正符号の符

号化がされる。光磁気ディスクを用いた記憶装置の場合、 $(M-P-N-Q=23)$ とされ、1個のマトリクスブロックが1個のセクターと対応する529バイトの大きさとされる。この529バイトの中の512バイトがデジタルデータとされ、他の17バイトがアドレス、識別コード、CRCコード等の付加データとされている。

$N$ 個の列ブロック $C1_1, C1_2, \dots, C1_N$ の夫々は、エラー訂正符号 $C1$ の符号系列であり、 $P$ 個のチェックシンボルを含んでいる。同様に、 $M$ 個の行ブロック $C2_1, C2_2, \dots, C2_M$ の夫々は、エラー訂正符号 $C2$ の符号系列であり、 $Q$ 個のチェックシンボルを含んでいる。つまり、列ブロック $C1_i$ を含む $Q$ 個の列ブロックは、符号 $C2$ のチェックシンボルに符号 $C1$ の符号化を行ったものであり、行ブロック $C2_j$ を含む $P$ 個の行ブロックは、符号 $C1$ のチェックシンボルに符号 $C2$ の符号化を行ったものである。エラー訂正符号 $C1$ 及び $C2$ としては、通常、線形符号が用いられる。例えば1シンボルエラーの訂正が可能

なリード・ソロモン符号がエラー訂正符号 $C1$ 及び $C2$ として用いられ、列ブロック及び行ブロックの夫々に $(P=Q=2)$ 個のチェックシンボルが含まれる。また、 $P$ 個の行ブロックと $Q$ 個の列ブロックとが重複する部分のチェックシンボルは、線形符号であるから行ブロック及び列ブロックの間で一致したものとなる。

第2図において破線で示すように、マトリクスブロックの斜め方向(対角線方向)に位置するシンボルの順序でデータが伝送される。エラー訂正符号 $C1$ 及び $C2$ の系列の方向と異なる斜め方向にデータを伝送するのは、伝送時に発生するバーストエラーをランダムエラーに分散化させ、エラー訂正符号 $C1$ 及び $C2$ によりエラー訂正が不可能となることを回避するためである。

$C1$ 復号器4及び $C2$ 復号器5は、第2図に示す種符号の復号を行う。メモリ1には、光磁気ディスクから再生されたマトリクスブロックの全てのデータが格納され、符号系列を形成する列ブロック又は行ブロック毎にメモリ1からデータが読

11

み出され、リード・ソロモン符号の復号がなされる。メモリ1からデジタルデータの列ブロック $C1_1, C1_2, \dots, C1_N$ が夫々読み出される時に、入出力制御回路3により、メモリ1及び $C1$ 復号器4の間が接続される。同様に、メモリ1からデジタルデータの行ブロック $C2_1, C2_2, \dots, C2_M$ が夫々読み出される時に、入出力制御回路3により、メモリ1及び $C2$ 復号器5の間が接続される。

リード・ソロモン符号の復号処理は、パリティ検査行列と各ブロックのシンボルとの乗算により2個のシンジローム $S_1$ 及び $S_2$ を求めるステップと、このシンジローム $S_1$ 及び $S_2$ からエラーの大きさをチェックするステップと、1シンボルエラーの時に、エラーを訂正するステップとからなる。

従来では、全ての列ブロック $C1_1, C1_2, \dots, C1_N$ についてのエラー訂正を行う $C1$ 復号と、全ての行ブロック $C2_1, C2_2, \dots, C2_M$ についてのエラー訂正を行う $C2$ 復号とを交互に繰り

12

返す方法で訂正処理がなされる。この発明は、従来の訂正処理と同等の訂正能力であって、より復号時間を短縮化できる方法であり、列ブロックの符号系列と行ブロックの符号系列とを1ブロック毎に交互に訂正するものである。第3図は、この発明の一実施例の訂正処理の方法のフローチャートである。第3図において、 $Y$ は肯定を表し、 $N$ は否定を表している。また、 $k$ は、列ブロックと行ブロックの訂正処理の順序を反転する毎に $(+1)$ されるカウンタであり、 $\alpha$ は、この反転処理の回数を制限するしきい値である。従って、 $(k \geq \alpha)$ の場合には、列ブロックと行ブロックの訂正処理の反転が行われない。

第3図に示されるように、復号処理は、二つのルーチン(A及びB)からなる。ルーチンAは、列ブロックの訂正処理を行い、次に行ブロックの訂正処理を行う順序の復号処理であり、ルーチンBは、行ブロックの訂正処理を行い、次に列ブロックの訂正処理を行う順序の復号処理である。ルーチンA又はルーチンBの一方の復号処理から開

13

14

始される。ルーチンAの復号処理について以下に説明する。

所定の列ブロックC1。及び所定の行ブロックC2。が復号開始の系列とされる。例えば(C1。=C1。)(C2。=C2。 )とされ、メモリ1からこれらのスタートブロックのシンボルが読み出される(ステップ①)。

次に列ブロックC1。の訂正処理及び列ブロックに関する1ビットのフラグF1が設定される(ステップ②)。列ブロックC1。の訂正処理において、エラーシンボルが無い時或いは1シンボルエラーが訂正される時に、(F1=G)と設定され、2個以上のエラーシンボルが在る時に、(F1=NG)と設定される。例えば(G="0") (NG="1")と定められている。

次に行ブロックC2。の訂正処理及び行ブロックに関する1ビットのフラグF2が設定される(ステップ③)。行ブロックC2。の訂正処理において、エラーシンボルが無い時或いは1シンボルエラーが訂正される時に、(F2=G)と設定

され、2個以上のエラーシンボルが在る時に、

(F2=NG)と設定される。

次にステップ④及び⑤において、フラグF1及びF2の状態の判定とカウンタkの値の検査がされる。(F1=NG)で且つ(F2=G)の条件が成立しない時(ステップ④)又はこの条件が成立しても(k<a)の条件が成立しない時(ステップ⑤)には、訂正終了信号が到来しているかどうかステップ⑥で調べられ、若しそうであれば訂正処理が終了する。

ステップ⑥において、訂正終了信号が到来していない時には、ブロックの番号(n及びm)が(+1)され(ステップ⑦)、ステップ①～⑤の処理が繰り返される。従って、ルーチンAにより、(C1。→C2。→C1。...→C2。...→C1。...)の順序で符号C1の系列と符号C2の系列とが1ブロック毎に交互に処理される。

他方のルーチンBは、上述のルーチンAの復号処理と同様のものである。つまり、所定のブロックのシンボルのメモリ1からの読み出し(ステッ

15

プ⑧)、行ブロックC2。の訂正処理及びフラグF2の設定(ステップ③)、列ブロックC1。の訂正処理及びフラグF1の設定(ステップ②)、フラグF1及びF2の状態の判定(ステップ④)、カウンタkの検査(ステップ⑤)が順次なされる。

(F1=G)で且つ(F2=NG)の条件が成立しない時又はこの条件が成立しても(k<a)の条件が成立しない時には、訂正終了信号が到来しているかどうかステップ⑥で調べられ、若しそうであれば訂正処理が終了する。

ステップ⑥において、訂正終了信号が到来していない時には、ブロックの番号(n及びm)が(+1)され(ステップ⑦)、ステップ①～⑤の処理が繰り返される。従って、ルーチンBにより、(C2。→C1。→C2。...→C1。...→C2。...)の順序で符号C1の系列と符号C2の系列とが1ブロック毎に交互に処理される。

ステップ④及び⑤において、(F1=NG)、(F2=G)、(k<a)の全ての条件が成立する場合には、カウンタkが(+1)され(ステッ

プ⑧)、他方のルーチンBのステップ③(列ブロックC1。の訂正処理及びフラグF1の設定)に移行する。従って、(C1。→C2。...)の訂正処理の次に列ブロックC1。の訂正処理が再びされる。これにより、[列ブロック→行ブロック]の順序が[行ブロック→列ブロック]の順序に反転される。

また、ステップ④及び⑤において、(F1=G)、(F2=NG)、(k<a)の全ての条件が成立する場合には、カウンタkが+1され(ステップ⑤)、他方のルーチンAのステップ③(行ブロックC2。の訂正処理及びフラグF2の設定)に移行する。従って、(C2。→C1。...)の訂正処理の次に行ブロックC2。の訂正処理が再びされる。これにより、[行ブロック→列ブロック]の順序が[列ブロック→行ブロック]の順序に反転される。

順序の反転は、(k<a)の時に、先に処理されるブロック(ルーチンAにおける列ブロック及びルーチンBにおける行ブロック)のフラグがN

17

18

Gで、後に処理されるブロックのフラグがGの状態の場合にのみなされる。また、復号終了時に、エラーシンボルが残っている時には、光磁気ディスクのドライブ制御装置にデータ再送要求が送出される。

上述の符号C1の系列と符号C2の系列とを1ブロック毎に交互に処理するこの発明によるエラー訂正は、第4図に示すようなエラーパターンの訂正に効果的である。

第4図は、簡単のため、マトリクスブロックが5行5列の積符号を示し、×で示すシンボルがエラーシンボルを表している。一般的な訂正処理として、1シンボルエラーの訂正が可能な符号C1の系列である列ブロックの全ての復号(C1復号)と1シンボルエラーの訂正が可能な符号C2の系列である行ブロックの全ての復号(C2復号)とを1回ずつ行くと、(C1復号→C2復号)又は(C2復号→C1復号)の何れの場合でも、破線で囲んだ4個のエラーシンボルが訂正されない。

19

できる。第4図に示すエラーパターンは、マトリクスブロックの斜め方向にデータを伝送する時にバーストエラーによって生じ易いものであり、バーストエラーの訂正にこの一実施例の訂正処理は、頗る有効である。

勿論、この発明は、積符号のみならず、マトリクスブロックの斜め方向にエラー訂正符号の符号化を行うもの等にも適用することができる。また、エラー訂正符号としては、リード・ソロモン符号以外の符号を使用でき、例えば1シンボルが1ビットの場合には、BCH符号を用いることができる。

#### 〔発明の効果〕

この発明に依れば、伝送時のバーストエラーにより、2つのエラー訂正符号C1及びC2の両者の系列に関して、複数個のエラーシンボルが含まれる系列が多いエラーパターン(C1復号及びC2復号を何回か繰り返すことにより最終的には訂正可能なエラーパターン)を復号する場合に、従

この一実施例では、列ブロックC1、又は行ブロックC2、の一方のブロック例えば列ブロックC1、がスタートブロックとされ、ルーチンAによって、(C1、→C2、→C1、→C2、)と訂正処理が順次なされる。この2番目の列ブロックC1、は、2個のエラーシンボルを含むので訂正不可能となり、フラグF1がNGとされる。この場合、行ブロックC2、がエラー訂正できるために、フラグF2がGとされる。従って、第3図におけるステップ④から、ステップ⑥を経てステップ⑦に移る。これによって、(C1、→C2、)の後に、列ブロックC1、の訂正処理が再びされる。このように、(C1、→C2、→C1、→C2、→C1、→C2、→C1、→C2、→C1、→C2、→C1、)の訂正処理によって、第4図に示す全てのエラーシンボルを訂正することができる。

つまり、訂正のステップ数が殆ど増加せずに、より多くのエラーシンボルを訂正することができ、限られた訂正処理の時間を有効に利用することが

20

来の復号方法に比して、少ないステップ数とすることができ、復号時間を短縮化することができる。

#### 4.図面の簡単な説明

第1図はこの発明の一実施例のブロック図、第2図はこの発明の一実施例の符号構成の説明に用いる略線図、第3図及び第4図はこの発明の一実施例の訂正処理の説明に用いるフローチャート及び略線図である。

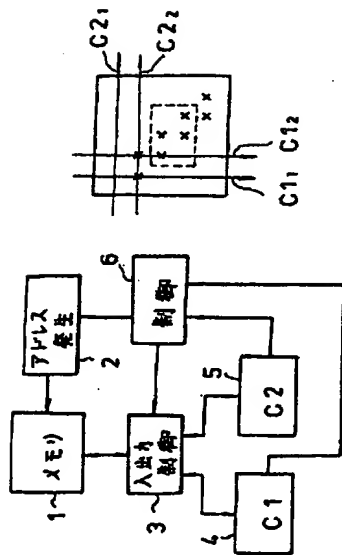
#### 図面における主要な符号の説明

1:メモリ、 2:アドレス発生回路、  
3:入出力制御回路、 4:C1復号器、  
5:C2復号器、 6:制御回路。

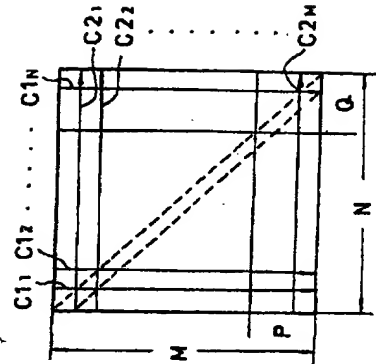
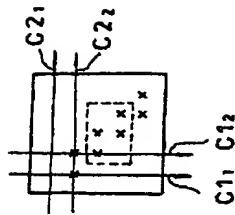
代理人 弁理士 杉 浦 正 知

21

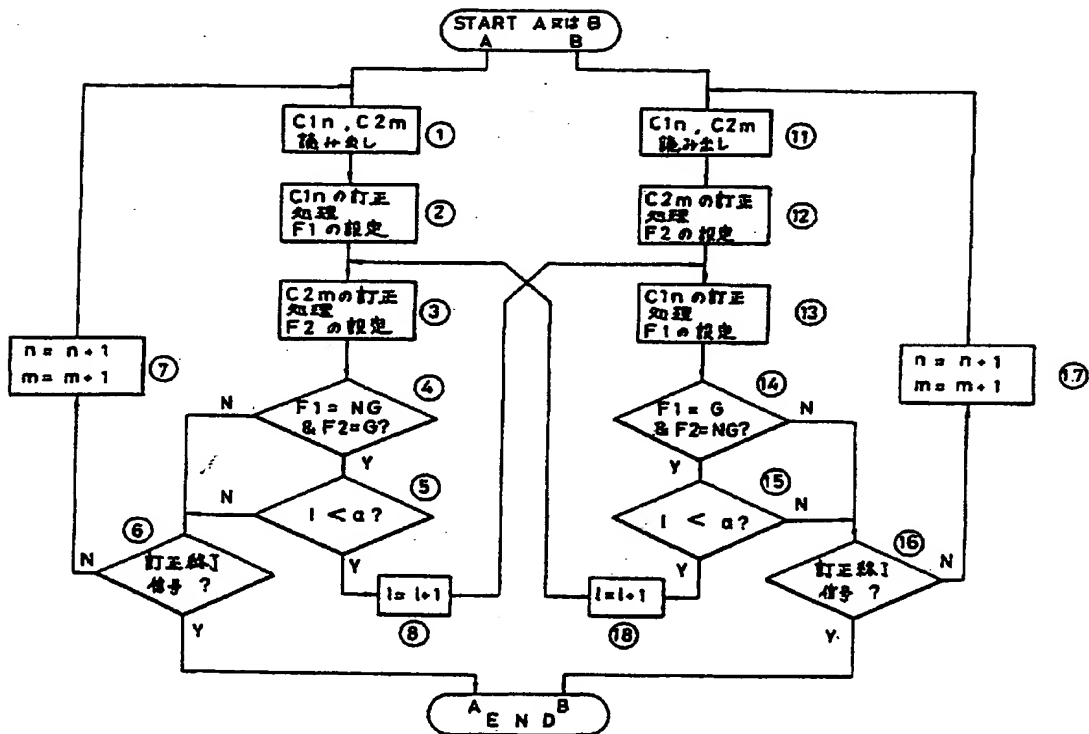
22



第1図  
70-70



第2図  
積符号



複号の70-4-4

第3図